



ISSN 1411 - 5972

# JURNAL Teknologi

(MAJALAH ILMIAH FAKULTAS TEKNIK - UNPAK)

Volume 1, Edisi 25, Periode Juli-Desember 2014

	Hal.
» Kata Pengantar	i
» Daftar Isi	ii
» Analisa Kebutuhan Modul Dan Baterai Pada Sistem Penerangan Jalan Umum (PJU) ( <i>Didik Notosudjono &amp; Asri</i> )	1
» Analisis Aspek Perizinan Pada Pembangunan Perumahan ( <i>Ike Pontiwaty</i> )	10
» Analisa Performasi Jaringan LAN (Lokal Area Network) IPTV ( <i>Yamato dan Evyta Wisniana</i> )	32
» Analisa Performa Virtualisasi Server Untuk Meningkatkan Efisiensi Data Center ( <i>Agustini Rodiah Machdi</i> )	42
» Kajian Bcton Berongga ( <i>Pervious Concrete</i> ) Sebagai Bahan Perkerasan Jalan Untuk Mengurangi Limpasan Air Permukaan ( <i>Titik Penta Artiningsih</i> )	53
» Analisa Perencanaan Jaringan Seluler CDMA 2000 1x (TELKOM <i>Flex</i> ) Untuk Layanan Voice Dan Data ( <i>Waryani</i> )	57
» Perencanaan Teknis Jalan Wilayah Selatan Provinsi Bengkulu Dengan Menggunakan Manual Desain Perkerasan Jalan (MDPJ) Nomor 02/M/BM/2013 ( <i>Arif Mudianto</i> )	75

# ANALISA PERFORMA VIRTUALISASI SERVER UNTUK MENINGKATKAN EFISIENSI DATA CENTER

Oleh :

**Agustini Rodiah Machdi**

## *Abstrak*

Virtualisasi adalah penciptaan sebuah versi virtual (bukan sebenarnya) suatu entitas, seperti sistem operasi, server, perangkat penyimpanan atau sumber daya jaringan. Alasan pemanfaatan teknologi virtualisasi adalah untuk memudahkan pengurangan administrasi dan biaya. Namun, implementasi saat monitor mesin virtual tidak memberikan isolasi kinerja yang cukup untuk menjamin efektivitas dari penggunaan sumber daya, terutama ketika aplikasi yang berjalan pada mesin virtual dari mesin fisik yang sama bersaing untuk komputasi dan komunikasi sumber daya. Dalam penelitian ini dijabarkan tentang studi pengukuran kinerja aplikasi jaringan I/O pada virtualisasi. Penelitian ini terfokus pada analisis berdasarkan pengukuran dampak kinerja alokasi aplikasi pada mesin virtual dalam hal throughput dan efektivitas berbagi sumber daya, termasuk dampak dari kasus host idle pada aplikasi yang berjalan secara bersamaan pada host fisik yang sama. Hasil penelitian ini menunjukkan bahwa dengan strategi pengalokasian aplikasi jaringan I/O dihasilkan peningkatan kinerja virtualisasi dari sisi pengguna mencapai angka 34%, dan dari sisi penyedia jasa virtualisasi dapat mencapai kinerja lebih dari 40%.

**Kata Kunci :** *Virtualisasi, virtual, VMWare, RAWC, kinerja virtualisasi, mesin virtual*

## 1. LATAR BELAKANG

Virtualisasi adalah penciptaan sebuah versi virtual (bukan sebenarnya) suatu entitas, seperti sistem operasi, server, perangkat penyimpanan atau sumber daya jaringan. Alasan pemanfaatan teknologi virtualisasi adalah untuk memudahkan pengurangan administrasi dan biaya. Pengurangan biaya berasal dari pengurangan jumlah server fisik, sehingga mengurangi daya listrik, kebutuhan sistem pendinginan server, dan pengurangan kebutuhan akan ekspansi ruangan. Daripada harus membeli hardware baru untuk mendukung aplikasi bisnis baru, dengan virtualisasi cukup hanya dengan menambahkan server virtual baru, gambaran mengenai virtualisasi server dapat dilihat pada gambar 1 berikut ini.



Gambar 1. Ilustrasi Virtualisasi Server

Namun penambahan virtual server baru tanpa mempertimbangkan pemanfaatan dan menghitung node yang ada akan menyebabkan peningkatan konsumsi energi dan sumber daya yang tidak diinginkan. Hal ini juga dapat meningkatkan penambahan infrastruktur fisik dan pendinginan peralatan yang pada akhirnya merupakan pemborosan energi.

Penelitian ini akan memfokuskan pada analisa performa virtualisasi server untuk mendapatkan manfaat yang maksimal dari teknologi virtualisasi server, aplikasi yang digunakan adalah VMware Desktop Reference Architecture Workload Simulator (RAWC).

RAWC digunakan untuk mensimulasikan beban kerja pengguna dalam lingkungan desktop Windows untuk melakukan stressing dan pembebanan pada media penyimpanan, jaringan dan beban kerja komputasi. RAWC mensimulasikan beban kerja yang biasa pengguna lakukan, seperti surfing internet, mengkompresi file, mengirim email, menggunakan aplikasi Microsoft Office dan Adobe Reader, dan sebagainya. Desktop

digunakan sebagai pengganti fisik dari server yang akan disimulasikan.

## 2. LANDASAN TEORI

Tujuan utama dari virtualisasi adalah untuk mensentralisasi tugas operasi dengan mengembangkan skalabilitas dan jumlah pekerjaan yang dapat dilaksanakan. Jadi intinya virtualisasi dijalankan dengan mengurangi resources seminim mungkin untuk dapat melakukan proses komputasi semaksimal mungkin.

Virtualisasi yang utama terbagi menjadi lima, yaitu :

- a) Virtualisasi hardware : virtualisasi yang melakukan eksekusi dari software dalam suatu area yang terpisah dari resources hardware. Jadi eksekusi dari suatu software dijalankan tidak berbasis pada keberadaan resources hardware secara fisik.
- b) Virtualisasi memory : virtualisasi dari perangkat memory dengan mengambil sebagian resources harddisk untuk menggantikan peran memory jika memory tidak mampu untuk menampung lagi.
- c) Virtualisasi storage : virtualisasi yang menyediakan resources storage yang secara logical ada namun secara fisik tidak ada.
- d) Virtualisasi data : virtualisasi dari data yang terbebas dari sistem database secara fisik, struktur dan storage-nya.
- e) Virtualisasi network : virtualisasi yang melakukan penggabungan resources network baik hardware dan software, dimana network berfungsi menjadi satu kesatuan yang secara administrative berbasis software, yaitu suatu virtual network

### 2.1. Virtual Desktop Infrastructure (VDI)

Virtual Desktop Infrastructure (VDI) adalah penerapan hosting sistem operasi desktop dalam mesin virtual (VM) yang berjalan pada server yang terpusat. VDI adalah variasi pada model komputasi client / server, kadang-kadang disebut sebagai komputasi berbasis server. Istilah VDI ini diciptakan oleh VMware Incorporated.

Virtual Desktop Infrastructure (VDI) terus mendorong ke arah yang lebih cost effective, aman, dan solusi terkelola untuk komputasi

desktop. Salah satu keuntungan VDI tersebut adalah pengguna remote desktop dapat terhubung langsung ke mesin virtual yang berjarak jauh di dalam sebuah data center. Sebuah faktor yang krusial pada penggunaan secara remote adalah infrastruktur hardware dan protokol remote display. Teknologi VDI mampu mengoptimalkan lingkungan virtual jarak jauh [4] untuk mewujudkan pengalaman pengguna yang lebih baik.

Evaluasi kinerja yang rinci dan studi infrastruktur hardware yang mendasari teknologi virtualisasi ini diperlukan untuk mengkarakterisasi penggunaan oleh end user. Studi subjektif sangat terbatas dan survei pada set data kecil yang tersedia yang menganalisis faktor-faktor ini [5,6,7] dan Hasil dan teknik tidak mengukur sampai skala persyaratan dalam VDI lingkungan. Sebagai contoh, jumlah yang diperlukan pengguna desktop dengan mudah berkisar dari beberapa ratus hingga puluhan ribu, tergantung pada jenis penyebaran. Oleh karena itu, ada kebutuhan mendesak untuk mekanisme otomatis untuk mengkarakterisasi dan rencana besar instalasi mesin virtual desktop yang. Proses ini harus secara kualitatif dan kuantitatif mengukur seberapa pengalaman pengguna bervariasi dengan besaran di lingkungan VDI.

Pengukuran ini penting memungkinkan administrator untuk membuat keputusan tentang bagaimana menyebarkan untuk hasil maksimal atas investasi (ROI) tanpa mengorbankan kualitas.

## 3. IMPLEMENTASI SISTEM VIRTUALISASI

Implementasi virtualisasi dilakukan pada sebuah PC dengan konfigurasi hardware Prosesor Intel Core i3-2120 3,3 Ghz , RAM 4 GB, Hardisk Seagate 500GB 7200 RPM SATA III dan koneksi jaringan 100Mbps. Sedangkan untuk konfigurasi software menggunakan VMware vSphere Hypervisor (ESXi), Microsoft Windows 7 dan VMware RAWC. Mesin fisik host beberapa virtual mesin. Setiap VM berjalan web server Apache untuk memproses permintaan web dari klien remote. Setiap klien menghasilkan file permintaan pencarian untuk mesin virtual tertentu sehingga klien tidak akan mengakibatkan hambatan. Setiap koneksi klien mengeluarkan satu permintaan file secara default. Sebuah node control klien

individual dan mengumpulkan data profil. Kinerja Web server diukur sebagai jumlah maksimum yang dapat dicapai dalam koneksi per detik ketika mengambil file dengan berbagai ukuran.

Kami menggunakan `httperf` [8] untuk mengirim permintaan klien untuk dokumen web berukuran 1kB, 10kB, 30kb, 50KB atau 70KB. Kriteria untuk file kecil atau besar tergantung pada kapasitas mesin. Untuk setup eksperimental ini file dengan ukuran yang lebih besar dari 10K dibatasi oleh jaringan.

#### 4. PENGUKURAN METRIK KINERJA VIRTUALISASI

Data dikumpulkan dengan menggunakan VMWare Desktop Reference Architecture Workload Simulator. Metrik yang digunakan dalam pengukuran penelitian ini adalah sebagai berikut :

1. Server Throughput (request/second), Ini merupakan data kuantitatif yang mengukur jumlah maksimum permintaan per detik yang sukses dilayani server ketika klien mengambil dokumen web
2. Normalized throughput.  
Dalam membandingkan throughput pada beberapa Mesin Virtual (VM) yang berbeda konfigurasi dan berbeda jumlah VM yang dimiliki dalam satu mesin fisik, biasanya dipilih satu pengukuran sebagai dasar referensi throughput dan normalized throughput dari beberapa setting konfigurasi yang berbeda dengan menggunakan rasio throughput yang diberikan oleh referensi dasar throughput dengan maksud untuk membuat perbandingan yang memadai.
3. Agregat throughput (req/second).  
Dalam penelitian ini digunakan gabungan throughput sebagai metrik untuk mengukur dampak penggunaan beberapa jumlah Mesin Virtual (VM) pada kinerja agregat throughput pada host secara fisik.
4. CPU time per execution (microsecond/exe).  
Ini adalah indikator kinerja yang menunjukkan rata-rata waktu yang diperoleh CPU dalam microsecond selama setiap eksekusi dijalankan pada domain.
5. Execution per second (exe/second).  
Metrik ini untuk mengukur jumlah guest domain yang dijadwalkan untuk dijalankan

pada CPU fisik dalam satu satuan waktu tertentu.

6. Utilisasi CPU (%).  
Untuk memahami pembagian sumber daya CPU di VMs yang berjalan pada mesin fisik tunggal, akan dilakukan pengukuran penggunaan CPU rata-rata setiap VM, termasuk penghitungan masing-masing penggunaan CPU Domain 0 dan CPU guest domain.
7. Network I/O (Input Output) per second (KByte/sec).  
Merupakan pengukuran jumlah lalu lintas pada I/O jaringan kB per detik, lalu lintas data yang ditransfer dari dan ke web server remote sesuai beban kerja.
8. Memori page exchange per second (pages/second).  
Merupakan pengukuran yang mengindikasikan seberapa efisien jumlah memori page exchange per detik pada channel I/O.
9. Memory pages exchange per execution (pages/exe).  
Ini adalah pengukuran memory page exchange per eksekusi (pages/exe) yang merupakan metrik indikator kinerja yang menunjukkan rata-rata halaman memori dipertukarkan selama setiap perintah dijalankan dalam satu domain.

##### 4.1. Idle Instance

Pada bagian ini akan dibahas secara mendetail analisa tentang kinerja VM yang idle instance, difokuskan pada biaya dan manfaat dari mempertahankan kondisi idle guest domain yang muncul sebagai beban kerja I/O pada VM yang terpisah tetapi dalam satu host fisik. Secara konkrit akan difokuskan pengukuran pada dua permasalahan berikut :

Mempelajari keuntungan dan kekurangan dari mempertahankan kondisi idle dari sudut pandang provider cloud dan pelanggan cloud. Mengukur dan memahami waktu yang diperlukan untuk memulai membuat satu atau lebih guest domain pada host fisik dan akibatnya pada aplikasi yang sudah ada. Dengan anggapan bahwa  $n$  ( $n > 0$ ) adalah jumlah VMs yang menginduk pada mesin fisik, guest domain (VM) mempunyai 3 kondisi yaitu :

Execution state, merupakan istilah untuk kondisi guest domain menggunakan CPU

Runnable state, kondisi dimana guest domain berada pada antrian CPU menunggu penjadwalan untuk dijalankan pada CPU  
 Blocked state, kondisi dimana guest domain di blok dan tidak berada dijalankan dalam antrian CPU.

Kondisi dimana guest domain disebut idle adalah pada saat guest OS nya dieksekusi dalam sirkulasi idle.

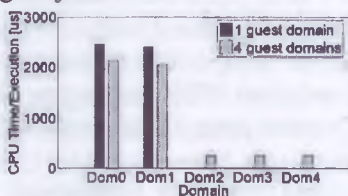
Tabel 1. Maksimum Throughput Domain

Besaran File Aplikasi	(jumlah guest domain, jumlah idle domain)			
	(1,0)	(2,1)	(3,2)	(4,3)
1 KB	1070	1067	1040	999
10 kB	720	717	714	711
30 kB	380	380	380	380
50 kb	230	230	230	230
70 kB	165	165	165	165

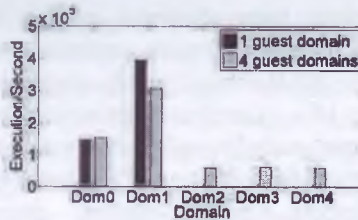
Tabel 1 merupakan hasil dari empat set percobaan yang dilakukan dimana Domain menjalankan aplikasi I/O dengan beban kerja tinggi pada nol, satu, dua dan tiga VMs dalam kondisi idle execution state.

Pada setiap set percobaan tercatat maksimum throughput yang dicapai untuk semua aplikasi I/O (1kB, 10kB, 30kB, 50kB dan 70kB).

Dari pengamatan percobaan pada table 1 terdapat beberapa fakta menarik. Pertama, terlihat dampak kinerja dipertahankan tetap berada pada kondisi domain idle ketika domain berjalan melayani aplikasi 30kb, 50KB dan 70KB, karena kinerja jaringan dibatasi. Kedua, penurunan kinerja terburuk terjadi pada aplikasi 1kB, yang sangat tergantung pada CPU. Dibandingkan dengan pengaturan VM tunggal di mana nilai throughput tertinggi yang dicapai adalah 1070 req/sec, juga terdapat penurunan kinerja sekitar 6% ketika jumlah guest domain adalah empat (999 req/sec). Hal ini jelas bahwa dengan menambahkan guest domain idle menyebabkan biaya overhead tinggi yang dapat mempengaruhi kinerja CPU secara intensif pada aplikasi dalam domain yang sedang berjalan.



Gambar 2. CPU time per execution ( $\mu\text{s}/\text{exe}$ ) untuk aplikasi 1 kB pada 1 VM dan 4 VM dengan 3 VM idle



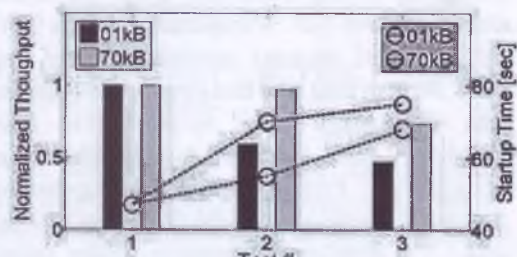
Gambar 3. Eksekusi per detik untuk aplikasi 1 kB pada 1 VM dan 4 VM dengan 3 VM idle

Gambar 2 dan Gambar 3 merupakan grafik untuk pengukuran yang lebih rinci mengenai dampak kinerja idle, yang membantu untuk mengukur secara kuantitatif overhead terjadi untuk aplikasi 1kB. Pengukuran waktu CPU per eksekusi sebagai jumlah eksekusi per detik dengan satu VM dan empat VMs dengan tiga setup siaga untuk aplikasi 1kB. Dari Gambar 2 dan Gambar 3, dihasilkan dua pengamatan. Pertama, pada masing-masing tiga guest domain yang idle didapat nilai rata-rata CPU time per execution  $250\mu\text{s}$  untuk masing-masing aplikasi yang berjalan, yaitu hanya sekitar 10% dari CPU time dari Domain0 per eksekusi. Kedua, membandingkan 4 VMs dengan 3 VMs idle yang di setup sebagai VM tunggal, terlihat CPU time untuk setiap yang dijalankan turun dari 2464 mikrodetik menjadi 2407 mikrodetik pada domain0 dan dari  $2130\mu\text{s}$  ke  $2046\mu\text{s}$  pada domain1, dan sama hasilnya dengan jumlah eksekusi per detik turun dari 400.000 sampai 300.000 di domain1, meskipun jumlah eksekusi per detik di domain0 terlihat ada sedikit peningkatan. Penurunan CPU time per execution dan eksekusi per detik (execution per second) ini terutama disebabkan dua faktor berikut:

- 1) Eksekusi timer tick untuk guest domain idle dan waktu pengalihan eksekusi antar domain, dan
- 2) Pengolahan paket jaringan seperti paket ARP (Advanced Resolution Protocol), yang menyebabkan terjadinya pengolahan I/O di guest domain.

Percobaan set kedua adalah untuk mempelajari seberapa intensif aplikasi CPU dan aplikasi I/O jaringan dalam mempengaruhi kinerja throughput pada saat sebuah instance idle muncul. Percobaan ini juga untuk mengetahui waktu startup yang dibutuhkan untuk membuat satu atau lebih guest domain baru dan faktor-faktor lain yang dapat mempengaruhi kinerja throughput seperti start-up time.

Metodologi yang diadopsi adalah sebagai berikut, Domain1 hanya melayani aplikasi 1kB atau 70KB kemudian dibuat satu atau dua instance idle. Gambar 4 mencatat fluktuasi throughput pada Domain1 dan waktu startup untuk guest domain yang idle. Grafik tersebut menunjukkan masing-masing throughput yang dijalankan pada Domain1 (exp1) secara tersendiri, Domain1 yang berjalan dengan startup satu VM on demand (exp2), dan Domain1 yang berjalan dengan startup dua VMs on demand (exp3).



Gambar 4. Throughput untuk aplikasi 1 kB dan 70kB dan startup time untuk satu dan dua guest domain baru

Dalam set percobaan ini, tingkat permintaan tetap pada 900 request/sec untuk aplikasi 1kB atau 150 request/sec untuk aplikasi 70KB, baik yang mendekati 90% dari nilai throughput maksimum yang terdapat dalam Tabel I. Sebagai dasar, sumbu utama y-axis adalah throughput yang dinormalisasi dengan tingkat keberhasilan 900 request/sec untuk aplikasi 1kB atau 150 request/sec untuk aplikasi 70KB. Sumbu y-axis kedua menunjukkan waktu start-up (detik) untuk satu, dua atau tiga VMs. Perhatikan bahwa lingkaran di exp1 menyatakan waktu startup untuk satu single-instance tanpa menjalankan Domain1. Gambar 4 menunjukkan tiga pengamatan yang menarik. Pertama, pada request start-up yang dijalankan oleh guest domain dengan jarak waktu yang pendek memiliki dampak yang buruk pada kinerja domain, apapun jenis aplikasi-host tersebut. Hal ini dikarenakan start-up I/O sangat intensif pada satu instance VM, yang berarti menciptakan satu instance sebesar 2GB pada guest domain. Sebagai hasil pengukuran menunjukkan bahwa untuk memulai sebuah VM baru, rata-rata konsumsi CPU adalah sekitar 20%. Konsumsi puncak CPU untuk menyelesaikan tugas ini bisa setinggi 75%. Selain itu, dibutuhkan sekitar 900 virtual blok untuk melakukan proses writing dan sekitar 200 virtual blok perangkat yang digunakan untuk proses writing. Kegiatan I/O yang

terkait untuk melakukan start-up domain baru tidak dapat diselesaikan tanpa adanya Domain0, yang memainkan peran penting dalam pengolahan beban kerja Domain1. Pengamatan kedua adalah bahwa aplikasi 70kb mengalami kekurangan dalam hal waktu start-up dari aplikasi 1kB. Hal ini karena kinerja aplikasi 70KB dibatasi jaringan dan mengkonsumsi lebih sedikit CPU, yang meredakan beban CPU. Dalam kasus ini aplikasi 1 kB akan mengkonsumsi sekitar 90% Sumber daya CPU di samping sekitar 5400 pertukaran memory page per detik antara Domain0 dan Domain1 untuk melayani 900 request/sec untuk aplikasi 1kB. Sebaliknya, hanya 60% Sumber daya CPU disediakan untuk melayani 150 permintaan / detik untuk aplikasi 70KB. Selanjutnya, untuk aplikasi 1kB, yang waktu startup yang dihasilkan dua guest domain di exp3 meningkat dari 47 detik pada exp1 menjadi 75 detik, yaitu sekitar 1,5 kali lebih besar. Sebaliknya, untuk aplikasi 70KB, perbedaan waktu start-up yang dihasilkan dua VMs ke satu VM relatif lebih kecil. Hal ini menunjukkan bahwa waktu start-up untuk membuat VMs baru sangat tergantung pada kedua jenis sumber daya aplikasi yaitu CPU dan Disk I/O yang di dijalankan domain dan jumlah VMs baru yang dibuat. Mengingat sumber daya CPU dan disk I/O terbatas dalam menciptakan domain baru, baik intensivitas CPU atau intensivitas disk I/O dan intensivitas aplikasi dalam menjalankan domain akan menghasilkan waktu start-up lebih lama daripada pembatasan aplikasi pada Network I/O. Sebagai pengamatan yang ketiga adalah bahwa durasi penurunan kinerja dialami karena dalam menciptakan VMs baru dalam percobaan ini dibatasi pada angka 100 detik. Berdasarkan percobaan ini durasi penurunan kinerja sangat tergantung kepada kapasitas mesin, tingkat beban kerja yang dijalankan pada domain, dan jumlah start up instance VM baru.

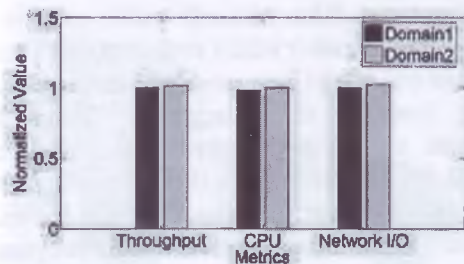
#### 4.2. Dampak Aplikasi Domain yang Berdampingan.

Dalam komputasi awan virtual, beberapa sumber daya seperti CPU, memori yang dibagi di beberapa VMs, sedangkan sumber daya lain seperti jaringan dan subsistem disk yang dibagi untuk beberapa VMs. Untuk itu dirancang tiga kelompok eksperimen untuk melakukan studi pengukuran dampak kinerja aplikasi co-locating dengan pola penggunaan

sumber daya yang berbeda dan jumlah VMs yang berbeda. Kelompok pertama dan kelompok kedua percobaan fokus pada dampak kinerja aplikasi yang dijalankan dengan pola penggunaan sumber daya yang berbeda. Untuk mengisolasi jumlah faktor yang berdampak pada dampak pola aplikasi co-locating, dipilih lima aplikasi I/O dalam percobaan ini yaitu 1kB, 10kB, 30kb, 50KB dan 70KB, tapi eksperimen dibagi menjadi dua langkah. Pada kelompok pertama, aplikasi yang identik dijalankan pada semua VMs untuk kelima aplikasi tersebut. Dilangkah kedua dipelajari skenario yang sedikit lebih kompleks di mana aplikasi yang berbeda yang berjalan pada VMs berbeda. Di kelompok percobaan ketiga, diteliti masalah distribusi beban kerja di antara beberapa VMs.

#### 4.2.1. Alokasi Aplikasi Identik

Dalam kelompok ini percobaan dirancang dua guest domain, Domain1 dan Domain2, baik melayani permintaan web identic menerbitkan pada tingkat beban kerja yang sama. Dalam skenario yang disederhanakan ini, Hasil penelitian menunjukkan bahwa ketika dua aplikasi I/O yang identik yang dijalankan secara bersamaan, urutan penjadwalan dibuat sama rata dalam berbagi CPU, konsumsi bandwidth jaringan, dan throughput yang dihasilkan.

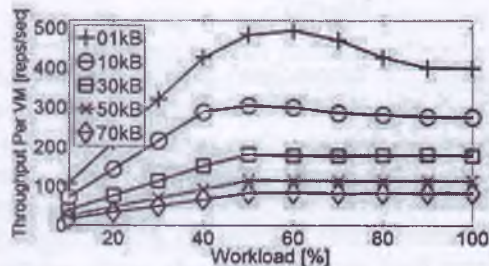


Gambar 5. Normalisasi throughput, utilisasi CPU dan I/O jaringan antara Domain1 dan Domain 2

Gambar 5 menunjukkan hasil eksperimen dua VMs saat keduanya melayani aplikasi 1kB dengan tingkat beban kerja 50%. Pengukuran throughput, utilisasi CPU, I/O jaringan yang didapat untuk Domain1 yang mengkonsumsi 36,1% sumber daya CPU sedangkan Domain2 mengkonsumsi 36,8% sumber daya CPU.

Para throughputs dan bandwidth jaringan masing-masing untuk domain1 dan Domain2

adalah: 480 req/sec (request per detik) dan 487 req/sec, 609 KByte/sec dan 622 KByte/sec. Terlihat tiga metrik nilai normalisasi yang menunjukkan kesamaan antar pengukuran. Untuk setiap pasangan metrik, kita menggunakan nilai untuk domain1 sebagai dasar perbandingan. Pada Gambar 5 perbedaan antara pengukuran di VM1 dan pengukuran VM2 adalah kecil dan dapat diabaikan.

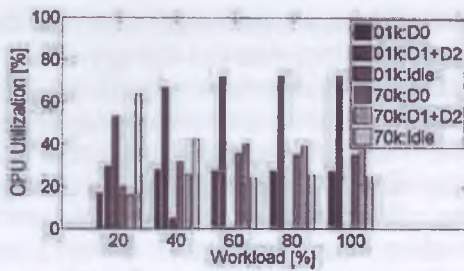


Gambar 6. Throughput rata-rata per domain (req/sec)

Gambar 6 merupakan grafik pengukuran throughput rata-rata Domain1 dan Domain2 untuk kelima aplikasi I/O. Hasil pengamatan didapat bahwa :

- 1) Semua aplikasi mencapai puncak kinerja pada saat diterapkan beban kerja 50% atau 60%,
- 2) Ada perbedaan penting antara file aplikasi berukuran kecil dan file yang berukuran besar aplikasi. Untuk file aplikasi berukuran kecil seperti 1kB dan 10kB, penurunan kinerja yang jelas dapat diamati pada tingkat beban kerja yang lebih tinggi dari 50% atau 60%. Namun, hal ini tidak terjadi pada kasus untuk aplikasi file yang berukuran besar. Secara signifikan lebih condong terjadi pada aplikasi 1kB karena:

- a. Kinerja dibatasi oleh sumber daya CPU,
- b. Guest domain menghabiskan lebih banyak waktu pada intensitas kedatangan paket jaringan ketika tingkat beban kerja tinggi,
- c. Dibandingkan dengan percobaan domain tunggal untuk kelima aplikasi yang ditunjukkan pada Tabel I, biaya overhead menjadi meningkat karena bridging jaringan terjadi di Domain0, dan switch jaringan.



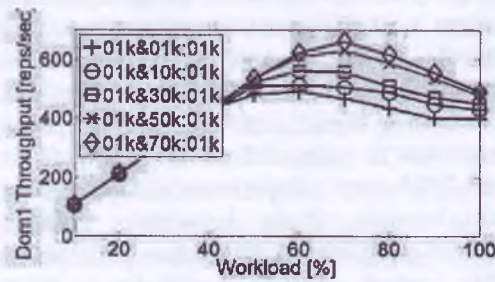
Gambar 7. Penggunaan CPU untuk Domain 0, penggunaan aggregate CPU untuk guest domain dan persentase CPU idle (%)

Gambar 7 merupakan grafik pengukuran penggunaan CPU untuk aplikasi 1kB dan 70kb pada tingkat beban kerja yang berbeda-beda. Pada pengukuran ini dilakukan penambahan penggunaan CPU secara bersamaan pada Domain1 dan Domain2 karena hasil yang didapat pada grafik gambar 5 menunjukkan bahwa Domain1 dan Domain2 selalu mencapai jumlah alokasi CPU yang sama. Gambar 7 menunjukkan di bawah tingkat beban kerja yang sama, penggunaan CPU guest domain untuk file 1kB jauh lebih besar dibandingkan dengan aplikasi 70KB, meskipun fakta bahwa nilai memory page exchange untuk file 1kB jauh lebih sedikit dibandingkan dengan aplikasi 70KB. Ini disebabkan karena CPU dikonsumsi untuk memproses permintaan jaringan yang terdiri dari dua komponen utama yaitu waktu yang dihabiskan untuk membangun koneksi TCP, dan waktu yang dihabiskan untuk mengangkut isi file web. Selain itu, koneksi fase secara signifikan membutuhkan lebih banyak sumber daya CPU pada fase transportasi.

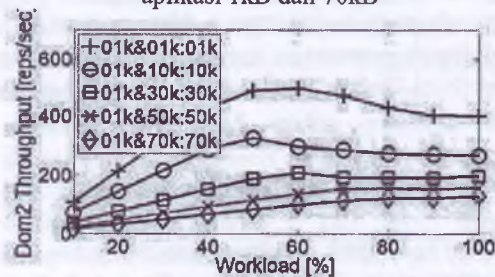
#### 4.2.2. Alokasi Aplikasi yang Berbeda

Dari hasil percobaan pada pokok bahasan sebelumnya, diketahui bahwa ketika dua aplikasi identic maka perkiraan nilai kelayakan dapat diperoleh dengan menggunakan default scheduler RAWC. Dengan demikian faktor utama yang mempengaruhi kinerja aplikasi alokasi pada mesin fisik yang sama diterapkan tarif beban kerja dan pola penggunaan sumber daya aplikasi. Dalam ayat ini kita akan mengkaji kinerja untuk domain tamu ketika mereka melayani aplikasi yang berbeda karena ini lebih mungkin terjadi dalam skenario dunia nyata. Kita mensimulasikan dua konsumen awan, ada yang menggunakan Domain1 dan melayani aplikasi 1kB, yang lain

menggunakan Domain2, menjalankan aplikasi, yaitu dengan desain yang bervariasi dari 1kB hingga 70KB.



Gambar 8. Throughput Domain1 saat melayani aplikasi 1 kB dan Domain2 melayani aplikasi 1kB dan 70kB

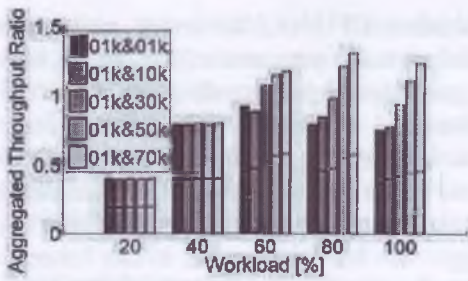


Gambar 9. Throughput Domain2 saat melayani Domain1 melayani aplikasi 1kB dan Domain2 melayani aplikasi 1kB dan 70kB

Gambar 8 dan Gambar 9 merupakan grafik pengukuran throughput untuk Domain1 dan Domain2 dengan beban kerja masing-masing dibawah 70%.

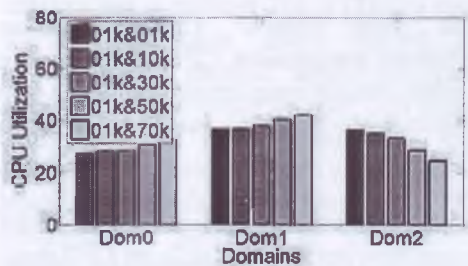
Terdapat dua fakta menarik yaitu, pertama, meskipun Domain1 selalu melayani file 1kB, kinerjanya sangat tergantung pada aplikasi yang berjalan di tetangganya Domain2. Sebagai contoh pada kombinasi 1kB dan 70KB (661 req/sec untuk 1kB) dibandingkan dengan kombinasi 1kB dan 1kB (494 req/sec untuk 1kB), nilai perbedaan kinerja yang didapat adalah 34%. Kedua, angka throughput tertinggi yang terjadi pada Gambar 8 dan Gambar 9 menunjukkan kecenderungan yang sangat berbeda. Sebagai contoh kombinasi aplikasi 1kB dan 70KB, untuk dua guest domain, angka throughput tertinggi muncul pada saat diterapkan beban kerja yang berbeda: titik tertinggi untuk file 1kB muncul pada tingkat beban kerja 70%, sementara itu beban kerja 100% muncul pada aplikasi 70KB. Jelas, fenomena ini terjadi karena pola penggunaan sumber daya aplikasi 1kB dan 70KB, aplikasi 1kB sangat terikat pada sumber daya CPU dan aplikasi 70KB sangat terikat pada sumber daya jaringan.





Gambar 10. Agregat rasio throughput untuk Domain1 dan Domain2 saat dibebani dengan 5 aplikasi

Gambar 10 adalah grafik hasil pengukuran rasio throughput yang dikumpulkan sebagai fungsi tingkat beban kerja. Penelitian ini menggunakan throughput maksimum VM tunggal selama lima aplikasi yang terdapat pada kolom pertama Tabel 1 untuk mendapatkan rasio throughput individu untuk setiap guest domain di bawah beban kerja yang spesifik. Sebagai contoh, throughput untuk Domain1 adalah 661 req/sec pada beban kerja 70%, sehingga rasio throughput sekitar 62% (yaitu, 661/1070). Selain itu didapatkan pula rasio throughput 130% untuk aplikasi 70KB. Dari hasil selama lima kombinasi aplikasi pada Gambar 10, diamati bahwa kasus co-locating yang terbaik adalah kombinasi 1kB dan 70KB dengan agregat rasio throughput 1,3, dan kasus terburuk adalah pada kombinasi aplikasi 1kB dan 1kB dengan nilai agregat seluruh rasio 0,92, perbedaan kinerja yang dicapai lebih dari 40% ( $(1.3-0.92)/0.92=41\%$ ).

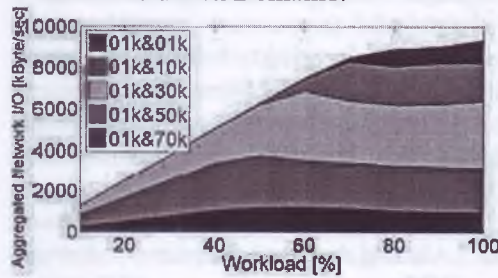


Gambar 11. Utilisasi CPU untuk Domain1 saat melayani 1kB dengan Domain2 melayani 1kB hingga 70kB dengan beban 100%

Gambar 11 grafik pengukuran penggunaan CPU dari Domain0, Domain1, dan Domain2 dari lima kombinasi aplikasi yang berbeda. Dibuat tiga pengamatan yang menarik dari set eksperimen ini :

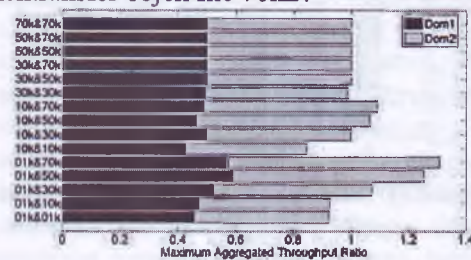
1. Diharapkan terjadi penurunan penggunaan CPU Domain2 ketika aplikasi di Domain 2 diubah ukurannya dari 1kB menjadi 70KB dan perubahan dari aplikasi CPU-intensive menjadi network-intensive.

2. Diharapkan terjadi peningkatan utilisasi CPU Domain0 ketika kombinasi aplikasi tetangga berubah dari kombinasi 1kB dan 1kB menjadi kombinasi 1kB dan 70KB, hal ini karena untuk aplikasi 70KB, jumlah besar data yang ditransfer menyebabkan konsumsi yang lebih tinggi dari driver perangkat di Domain0.
3. Diharapkan terjadi peningkatan utilisasi CPU Domain1 karena itu dapat menjelaskan mengenai peningkatan kinerja yang terjadi untuk aplikasi 1kB seperti yang terlihat pada pada Gambar 8, ketika domain2 berubah berpasangan dengan aplikasi jaringan 70KB, sehingga memungkinkan terjadinya pelepasan sumber daya CPU di domain2 yang dimanfaatkan oleh aplikasi 1kB CPU-intensive dalam Domain1.



Gambar 12. Agregat I/O jaringan saat Domain1 melayani aplikasi 1kB dan Domain2 melayani aplikasi 1kB hingga 70kB

Gambar 12 merupakan grafik plot agregat I/O jaringan yang konsumsi sebagai fungsi dari tingkat beban kerja pada domain tetangga dengan lima kombinasi aplikasi yang berbeda. Dari hasil pengamatan bahwa kombinasi aplikasi 1kB dan 70kb mengkonsumsi bandwidth jaringan tertinggi. Hasil eksperimen ini konsisten dengan hasil pada Gambar 10 dan Gambar 11. Untuk kombinasi aplikasi 1kB dan 70kb, sedangkan sebagian besar (80%) permintaan diproses pada saat permintaan pengambilan file 1kB (lihat utilisasi CPU domain1 untuk kombinasi aplikasi 1kB dan 70KB), sebagian besar byte jaringan (94%) dikonsumsi pada saat mentransfer objek file 70kB.



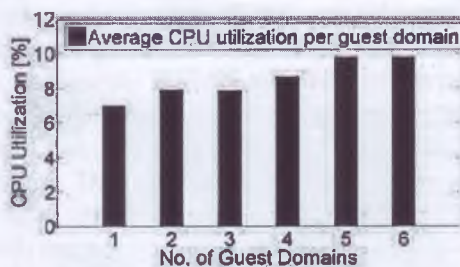
Gambar 13. Agregat maksimum rasio throughput untuk semua kombinasi guest domain

Gambar 13 menunjukkan grafik maksimum agregat rasio throughput untuk semua kemungkinan pada kombinasi dua domain tamu dan lima ukuran file aplikasi yang berbeda. Tujuannya adalah untuk dapat lebih komprehensif meneliti dampak kinerja dari berbagai jenis kombinasi aplikasi. Ini adalah sebuah percobaan super set dari Gambar 10 di mana lima kombinasi diperiksa. Didapat tiga pandangan dari grafik pada gambar 13 ini yaitu :

- 1) Aplikasi intensif dua CPU, seperti kombinasi 1kB dan 10kB maka kinerja kedua aplikasi akan merosot jauh.
- 2) Jika dipilih kombinasi aplikasi jaringan intensif seperti 30KB dan 50KB, maka setiap aplikasi akan memberikan kontribusi kinerja sama yaitu 50% dari agregat throughput, dan tidak ada kinerja degradasi.
- 3) Strategi co-locating terbaik adalah dengan menggunakan CPU intensif dan jaringan kombinasi aplikasi intensif, yang selalu mencapai throughput yang dikumpulkan mencolok tinggi.

#### 4.2.3. Alokasi Aplikasi Diantara Beberapa VMs

Setelah diteliti dampak dari alokasi aplikasi di dua guest domain di node host fisik tunggal. Pada bagian ini dikhususkan pada studi pengukuran mengenai dampak alokasi aplikasi pada beberapa VMs. Set percobaan pertama dirancang dengan memvariasikan jumlah guest domain dari satu sampai enam dan masing-masing guest domain diterapkan beban kerja sebesar 10%. Total tingkat beban kerja dapat dihitung dengan mengalikan jumlah domain tamu dengan tingkat beban kerja yang diterapkan. Menggunakan 10% beban kerja yang diterapkan untuk setiap guest domain untuk menjamin agar tidak terjadi pertentangan pemakaian sumber daya.

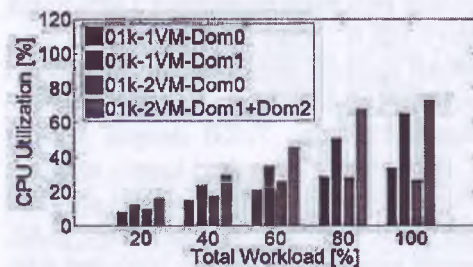


Gambar 14. Utilisasi rata-rata CPU untuk setiap VM yang masing-masing diberi beban 10%

Gambar 14 menunjukkan ketika ada enam guest domain yang berjalan, waktu yang

dihabiskan CPU (9,8%) untuk memproses jumlah data I/O yang sama (10% beban kerja per guest domain) setara dengan 1,5 kali dari waktu CPU dihabiskan (6,9%) dalam kasus guest domain tunggal. Kelompok eksperimen ini bermaksud untuk menunjukkan perbandingan antara kasus guest domain tunggal dan banyak domain, ketika beberapa guets domain berjalan konteks beban kerja beralih di antara beberapa guest domain sehingga lebih sering menyebabkan cache miss dan TLB miss [8], yang akan menghasilkan lebih banyak konsumsi waktu CPU dalam melayani data yang sama. Konteks sumber daya yang terpakai pada peralihan antar VM biasanya sebanding dengan jumlah domain tamu host pada mesin fisik.

Untuk percobaan set kedua, total tingkat beban kerja diubah menjadi 20%, 40%, 60%, 80% dan 100%. Digunakan aplikasi 1kB dan aplikasi 70KB karena kedua aplikasi ini merupakan dua aplikasi yang representatif.



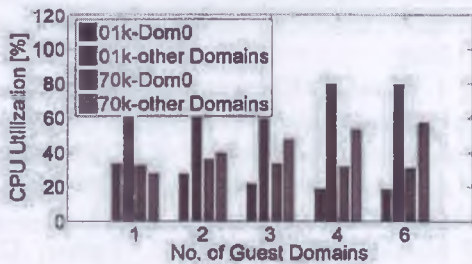
Gambar 15. Penggunaan CPU oleh satu dan dua guest domain dengan beban kerja bervariasi

Grafik pada gambar 15 menunjukkan pengukuran penggunaan CPU untuk domain dan guest domain yang berada dalam dua jenis konfigurasi mesin virtual yaitu : VM tunggal dan dua VMs.

Sebagai contoh, 01k-1VM-Dom0 menunjukkan pengukuran utilisasi CPU dom0 untuk aplikasi 1kB yang berjalan pada VM tunggal. 01k-2VM-Dom0 menunjukkan pengukuran utilisasi CPU dom0 untuk aplikasi 1kB yang berjalan pada dua VMs. 01k-2VMDom1 + Dom2 mengukur penggunaan CPU gabungan dari Domain1 dan Domain2 untuk aplikasi 1kB. Dua VCPU (Virtual CPU) dikonfigurasi untuk setiap guest domain. Ketika dua guest domain berjalan, enam VCPUs menunggu untuk dijadwalkan berjalan di dalam CPU fisik, dibandingkan dengan empat VCPU dalam kasus satu guest domain. Terjadinya konteks peralihan sumber daya CPU dikenakan tidak

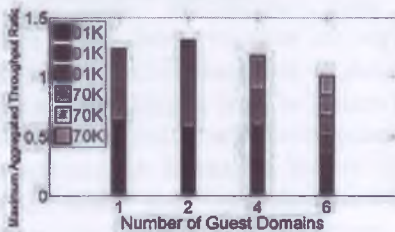
diinginkan terjadinya cache miss dan TLB miss. Pada percobaan dua guest domain, Domain0 harus berurusan dengan konteks peralihan sumber daya dan penjadwalan tambahan, juga proses bridging jaringan tambahan karena harus mentransfer paket ke guest domain individu.

Jadi Domain0 mendapat bagian sumber daya CPU yang lebih besar karena digunakan untuk pengaturan dua guest domain. Hasil set percobaan penelitian ini juga menunjukkan bahwa penggunaan CPU oleh guest domain meningkat seiring dengan tingkat beban kerja sehingga mendekati angka 100%.



Gambar 16. Penggunaan CPU oleh satu, dua, tiga, empat dan enam guest domain dengan beban kerja 120%

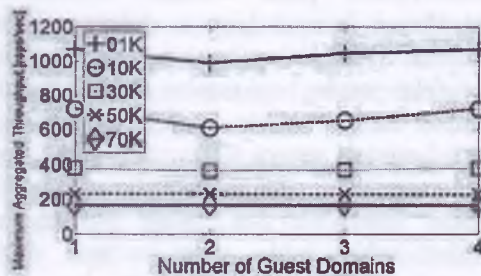
Grafik pada gambar 16 menunjukkan penggunaan CPU dalam situasi pertentangan tinggi. Variasi total tingkat beban kerja mencapai angka 120%. Seperti yang terlihat pada gambar 16, ketika jumlah guest domain meningkat dari satu sampai enam, pembagian CPU untuk Domain0 berkurang secara bertahap untuk aplikasi 70KB (32,5% sampai 31,3%). Sebaliknya, ketika jumlah guest domain diubah menjadi enam, penggunaan CPU di Domain0 untuk aplikasi 1kB menjadi berkurang dari 33,8% menjadi 18,6%. Untuk aplikasi 1kB, pengurangan yang signifikan dalam pemanfaatan Domain0 CPU menunjukkan pertentangan CPU tumbuh karena pertumbuhan yang berkelanjutan dalam penggunaan CPU guest domain. Urut-urutan penjadwalan dimaksudkan untuk membagi rata sumber daya CPU diantara domain termasuk Domain0.



Gambar 17. Agregat rasio throughput untuk satu, dua, tiga, empat dan enam VM, sebagian melayani aplikasi 1kB dan sebagian lainnya melayani aplikasi 70kB

Gambar 17 menunjukkan grafik pengukuran dampak jumlah guest domain terhadap agregat rasio throughput dengan setengah dari guest domain yang melayani aplikasi 1kB dan setengah lainnya yang melayani aplikasi 70KB. Penggunaan throughput maksimum pada VM tunggal untuk lima aplikasi di kolom pertama dari Tabel 1 sebagai dasar untuk mendapatkan rasio throughput individu untuk setiap guest domain.

Pada pengamatan pertama dilakukan setting konfigurasi untuk dua domain tamu melebihi konfigurasi domain lainnya lainnya. Ketika jumlah domain tamu beralih ke empat atau enam, agregat rasio throughput berkurang angkanya menjadi satu. Ini berarti ketika beberapa guest domain konfigurasinya diseting melampaui beberapa nilai konfigurasi domain lainnya, yaitu empat dalam kasus ini, tambahan sumber daya meningkat oleh hosting yang memiliki beberapa guest domain sehingga menguras habis manfaat dari kombinasi konfigurasi terbaik. Dibandingkan dengan kasus enam guest domain (1,01), kasus terbaik (1,31) ketika ada dua guest domain menunjukkan peningkatan kinerja sekitar 30%.



Gambar 18. Throughput untuk satu, dua, tiga dan empat guest domain yang menjalankan lima set aplikasi

Grafik pada gambar 18 merupakan pengukuran kinerja keseluruhan throughput maksimum untuk lima aplikasi, yang masing-masing didistribusikan pada beberapa guest domain. Set percobaan ini menunjukkan bagaimana karakteristik aplikasi yang dikombinasikan dengan jumlah guest domain tamu secara bersama-sama dapat mempengaruhi kinerja throughput keseluruhan. Untuk aplikasi 10kB, throughput gabungan maksimum adalah 608 req/sec pada saat dua guest domain berjalan, dibandingkan dengan angka throughput gabungan maksimum 720 req/sec yang dicapai dalam percobaan satu guest domain, yang berarti menunjukkan penurunan kinerja sebesar 15%.

Pada saat tiga guest domain dimunculkan, throughput agregat adalah 652 req/sec. Bila dibandingkan dengan kasus guest domain tunggal, penurunan kinerja yang terjadi adalah 10%. Terjadi sedikit peningkatan throughput untuk kasus empat guest domain hal ini dikarenakan kemampuan load balancing pada urutan-urutan global scheduler [9].

## 5. KESIMPULAN

Untuk memaksimalkan manfaat dan efektivitas konsolidasi server dan konsolidasi aplikasi dalam lingkungan virtualisasi, karenanya penting untuk melakukan pengukuran kinerja yang mendalam untuk aplikasi yang berjalan pada beberapa VMs yang menginduk pada mesin fisik tunggal. Pengukuran tersebut dapat memberikan analisa baik secara kuantitatif dan kualitatif mengenai lambatnya kinerja pada lingkungan virtualisasi, untuk mendapatkan pemahaman yang lebih dalam adalah merupakan faktor kunci untuk berbagi sumber daya yang efektif di antara aplikasi yang berjalan di lingkungan virtualisasi. Penelitian ini menelaah studi pengukuran kinerja aplikasi I/O jaringan dalam lingkungan virtualisasi. Dengan memfokuskan pada analisis berdasarkan pengukuran dampak kinerja alokasi aplikasi di mesin virtual dalam hal kinerja throughput dan efektivitas berbagi sumber daya, termasuk dampak dari kasus idle domain pada aplikasi yang berjalan secara bersamaan pada host fisik yang sama, dan perbedaan penjadwalan sumber daya CPU dan strategi alokasi, dan tingkat beban kerja yang berbeda dapat mempengaruhi kinerja sistem virtual.

Dalam penelitian ini, didapat hasil bahwa dengan strategi alokasi aplikasi jaringan I/O secara bersama-sama, diperoleh keuntungan kinerja yang cukup tinggi.

## DAFTAR PUSTAKA

[1] Sugiri, Haris S, VMware Solusi Menggunakan Beberapa Sistem Operasi, Andi Publisher, Yogyakarta 2010

- [2] Ir. Hendra Wijaya, VMware Workstation, Elex Media Komputindo, Jakarta 2013.
- [3] Iwan Sofana, Teori dan Praktik Cloud Computing, Elex Media Komputindo, Jakarta Januari 2013
- [4] Menon, A. L. Cox, W. Zwaenepoel, Optimizing Network Virtualization in Xen, 2006 USENIX Annual Technical Conference.
- [5] K. K. Ram, J. R. Santos, Y. Turner, A. L. Cox, S. Rixner, Achieving 10 Gb/s using Safe and Transparent Network Interface Virtualization, VEE 09.
- [6] G. Somani and S. Chaudhary, Application Performance Isolation in Virtualization, IEEE Int. Conf. on Cloud Computing, 2009.
- [7] T. Wood, L. Cherkasova, K. Ozonat, and Prashant Shenoy, Profiling and Modeling Resource Usage of Virtualized Applications, Middleware 2008, LNCS 5346, 2008.
- [8] Menon, J.R. Santos, Y. Turner, G.J. Janakiraman, and W. Zwaenepoel, Diagnosing Performance Overheads in the Xen Virtual Machine Environment, ACM/USENIX International Conference on Virtual Execution Environments, VEE 05, 2005.
- [9] L. Cherkasova, D. Gupta, A. Vahdat, Comparison of the Three CPU Schedulers in Xen, ACM SIGMETRICS Performance Evaluation Review, Vol. 35, Issue 2, September 2007.

## PENULIS :

*Agustini Rodiah Machdi, ST., MT.* Staf Dosen Program Studi Teknik Elektro, Fakultas Teknik – Universitas Pakuan, Bogor